Dynamic Routing for Multi-Homed Enterprise Gateways

Alexander A. Kist
University of Southern Queensland
Toowoomba, Queensland 4350, Australia
Email: kist@ieee.org

***Abstract*— Small and medium sized businesses often operate local area networks that connect via service providers to the Internet. If multiple connections are required, there are two options available: A gateway router that operates the border gateway protocol or a simpler gateway appliance that mostly uses load balancing and often network address translation. However, there is no simple way to multi-home small networks, when access links have diverse performance parameters, capacities or costs. Dynamic overflow routing addresses this issue in a straightforward way. This paper discusses dynamic routing and details, how bandwidth resources can be managed. It gives a simple estimate that helps to forecast resources for routing decisions and it underlines the need for a bidirectional gateway routing mechanism, accounting simultaneously for upstream as well as downstream traffic.**

I. Introduction

Traditionally, IP network routing is not sensitive to network traffic loads; it is mostly static in between route updates. Routes are selected by protocols and these are used until new routes are calculated. Routing protocols can react dynamically, for example to link failure, but generally they won‘t react to traffic load shifts and changes in the demand. These protocols are concerned with connectivity. Resulting issues and short comings have been acknowledged by the research community for a long time: in the case of incorrectly dimensioned resources, changes in network traffic or equipment failures, certain links in the network may become congested whereas other network parts may be underutilised.

Traffic Engineering methods for intra domain IP networks have been widely studied, including methods that allow arbitrary routing, for example *Multiprotocol Label Switching* (MPLS) [1] and methods that redistribute load such as *Open Shortest Path* (OSPF) optimisation (e.g. [2]). The *Border Gateway Protocol* (BGP) is the de facto inter-domain routing standard. BGP provides certain rules that influence traffic flows and their routing e.g. ([3], [4]). However, these schemes are not load sensitive and most of the methods rely on the manipulation of prefix matching and routing tables. Inbound traffic routing depends mostly on policies that are implemented by upstream ISPs. Dynamic routing schemes have been used in public switched telephone networks for a long time. Examples include Dynamic non-hierarchical routing (DNHR) [5] which uses different path sets for different times of the day, *Dynamically Controlled Routing* (DCR) [6], *Dynamic Alternative Routing* (DAR) [7] and *State- and Time-Dependent Routing* (STR) [8].

However, dynamic, load aware routing as such has not been popular in IP networks. Besides others, two issues can be identified: Packet reordering and stability. If packets that belong to the same flow are routed on different paths, packets can arrive out of order at the destination. This has a negative effect on TCP as well as many applications. This has been addressed by the *Scheme for Alternative Packet Overflow Routing* (SAPOR) [9]. SAPOR enables overflow routing in IP networks and routes traffic on a flow bases. If current routing protocols are used to propagate resource information, frequent route changes (route flapping) can be observed. A dynamic routing implementation has to account for these issues. Packet reordering can be addressed by flow based routing, i.e. all packets that belong to the same flow are routed on the same interface. Stability issues relate to available bandwidth estimation and the propagation of this information. This is specific to the investigated scenario and the routing protocols used. In this paper, the second issue is addressed and an improved method for bandwidth estimation is proposed.

Furthermore, the focus is on gateway routing: One network domain is connected to several upstream *Internet Service Providers* (ISPs). In this scenario the propagation of load information is not an issue. The main factor here is bandwidth estimation, i.e. to determine how many additional flows can be accommodated by links. Larger networks usually use BGP capable routers to interface with the Internet, smaller networks do not employ intelligent routing protocols but use gateway appliances to route traffic to the Internet. May networks also use network address translation (NAT). For multiple connections two routing strategies are widely used: load balancing and failover. In many ways these strategies are not optimal. Better results can be provided by dynamic, load and performance sensitive routing.

The contributions of this paper are threefold: Firstly, it introduces a dynamic routing application that allows multi-homing for networks that do not use inter-domain routing protocols and operate in a NATed environment. Secondly, it shows a simple model that allows resource forecasts for a given uncertainty. Thirdly, it outlines the need, for routing to account for load in both directions and it gives exemplary simulation-results that demonstrate the advantages of overflow routing. Section II details the gateway scenario and related problems that are the focus of this discussions, Section III introduces the enterprise overflow router as a prototype implementation for dynamic routing, Section IV, discusses

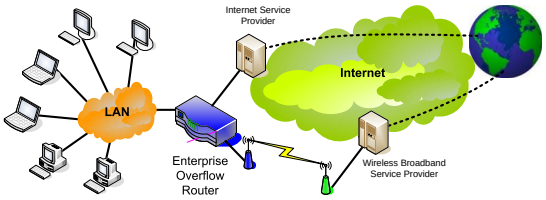

Fig. 1. Multi-Homed Network Wire Line and Wireless

bandwidth estimation, Section V assesses routing strategies and Section VI provides simulation results.

II. Scenarios, Motivation and Assumptions

Small and medium size networks are often connected via gateways. If more than one connection is required, load balancing is commonly used in multi-homed networks. Larger Networks use BGP as a gateway routing protocol. This paper focuses on the case where the network does not use the border gateway protocol and WAN interfaces are NATed. If both connections are not equal, standard load balancing or failover techniques are not optimal. Variations can include mixed access technologies, different equipment, different upstream/downstream capacity, and different performance parameters.

In a standard gateway setup, one or more public IP addresses are translated into a private address pool. To use more than one connection in such a scenario and additional public IP addresses, separate network address translation (NAT) is required for each outgoing interface. This has the additional advantage that traffic that is sent on one interface will also cause return traffic on the same interface.

The investigated scenario can be summarised as follows: the router has one local network connection (lan) and two or more external network connections (wan). Figure 1 depicts that basic scenario. The cooperate LAN is connected to two ISPs.

The motivation is to apply the concept of overflow routing in the above presented scenario. This will allow the simultaneous use of multiple links that have different metrics, e.g. cost or performance. If the interfaces can be prioritised by a metric and requested destinations are available via both links, overflow routing is the optimal strategy. It will use the best (cheapest, shortest delay, etc.) link first and only if necessary will it use the other links.

For the overflow algorithm practical estimates are required. The underlying assumptions for the remainder of this paper can be summarised as follows:

1) Each WAN interface uses NAT: This implies that all return traffic is routed on the same interface than emanating traffic.

2) Public IP addresses are bound to one interface: It follows that traffic that is initiated on the WAN side (for local servers) is tied to the interface where the request is received.

3) Flow rates can adapt: TCP flows can adapt their rate to the available link capacity.

4) Flow rates are static for one sampling interval: Flow rates can be modelled as random variables drawn from a finite distribution with (known) mean and standard deviation.

III. Enterprise Overflow Router

The Enterprise Overflow Router is a flow based router prototype implementation. Its mechanism will be briefly explained. The router enables flow based routing and implements three principles: Firstly, it makes sure that packets that belong to the same flow are routed on the same interface, also in the overflow case. Secondly, the number of additional flows that can be accommodated is determined. And thirdly, if the target bandwidth is reached, additional flows are routed on alternative interfaces. A hash based flow tracker implements the first principle; the second and third principles are implemented by a token system.

The enterprise overflow router operation can be summaries as follows: It is determined if arriving packets are part of a flow that is already tracked. If the flow exists the packet is marked for transmission on the interface that is associated with this flow. The traffic is added to the link traffic measure and the packet is forwarded using the interface mark.

If an unknown flow arrives, it is added. In the overflow case, it has to be determined if the flow can be accommodated on the default interface: Are tokens available in the forward and reverse direction? If the default link is available, the flow is routed on the default link, the packet is marked with the interface and the interface flow count is increased. The traffic is recorded and the packet is forwarded. If the default link is unavailable, overflow links are checked for available resources in the order of preference. If no vacant overflow links are found, the flow is routed on the default link (or an interface is determined randomly). If an overflow link is available, the overflow interface flow count is increased, the traffic is added and the packet is forwarded on the overflow link.

Stale flows have to be removed from the table of active flows. If no more packets are received that belong to a given flow within a preset time interval (e.g. 1 sec), the flow is cleared and the corresponding token is returned to the link budget. The available capacity on the links is determined by a token system. Details are discussed in Section IV. At defined time intervals available resources (tokens) are updated.

One main concern of flow based router implementations in the past was the scalability of the approach. However, processing and memory requirements for each flow are minimal and complex tasks can be executed in longer time intervals. The prototype is implemented on an embedded system with a Celeron M 600Mhz. The implementation is not optimised yet, but the system can comfortably handle traffic at line speed (100 Mbits) at a CPU load of 20% to 70%.

IV. Bandwidth Estimation Models

Flow models that are implemented in the overflow router have to be fairly simple to allow real-time online calculations of available bandwidth. Furthermore, dynamic routing requires a simple way of judging and propagating available resource information. New flows are constantly arriving and old flows are timing out and the flow rates of some flows adapt. To manage this dynamic behaviour resources are evaluated in given time intervals. The routing mechanism uses tokens to manage bandwidth. Two implementations have been proposed: tokens can account for the current average flow rate or they represent a fixed bandwidth fraction. In the first case, one token is assigned per flow, in the second, multiple tokens can be assigned to a flow. The underlying aim is to be able to judge how many additional flows can be accommodated by a link; therefore to determine how many tokens are assigned to the link budget.

A. One Token per Flow

Tokens represent the number of flows that have been active in the last measurement interval. The used bandwidth is recorded for this interval. Using bandwidth and token count the average bandwidth for the last sampling interval can be calculated. Earlier work [10] has developed a blocking model that can estimate flow blocking probabilities if the flow rate distributions mean and standard deviations are known. The model provides accurate results in the case where the link load is below the maximum link utilisation. This case applies: To be able to route additional flows, the load has to be below the maximum utilisation. A similar approach can be used to evaluate the usefulness of the flow rate mean to predict the number of additional flows that can be accommodated by a link.

For the following calculations it is assumed that the mean and standard deviation of the flow rate distribution are known. Both can be easily measured in practise if required. The mean flow rate determines the number of flows that can be accommodated by a link of capacity C. The selection of i flows from the flow rate distribution is equivalent to drawing i samples from a probability distribution with mean μ_f and standard deviation σ_f. To find the sampling distribution of the mean flow size, the central limit theorem can be applied: The sample mean $\overline{y}$ is approximately normally distributed with mean $\mu_{\overline{y}} = \mu_f$ and standard deviation of $\sigma_{\overline{y}} = \frac{\sigma_f}{\sqrt{i}}$. The coefficient of variation (c_v) is used as a measure of dispersion for the flow rate distribution. Equation (1) shows the calculation of the upper bound of the mean (μ_+) for about 97.7% confidence that the mean lies below this value. This is equivalent to a normal distribution between $(\infty, 2\sigma]$.

$$\mu_+ = \mu_f \left(1 + \frac{2c_v}{\sqrt{i}}\right) \tag{1}$$

The variable i indicates the number of flows that are currently active. For example, if a link accommodates 20 flows and the c_v of the distribution equals two, $\mu_+ = 1.89\mu$. For random traffic ($c_v = 1$) and 20 flows $\mu_+ = 1.44\mu$. The average flow rate μ allows to estimate the number of additional flows the link can accommodate, shown in Equation (2).

$$N = \frac{C}{\mu} - i \tag{2}$$

The number of additional flows N depends on the choice of μ. A conservative estimate is given by μ_+. N is set as the link budget for dynamic routing. It is decreased for every new flow and increased for each flow that times-out. N is revaluated for every measurement interval. Since the future flow sizes are unknown, the underlying assumption is that flow rates of flows arriving within the next interval have a similar statistical distribution than current flows. Keeping in mind that for this application the main interest is a practical number of flows that can be accommodated, this appears to be a valid assumption.

B. Fixed Bandwidth Fraction Tokens

An alternative way to account for used bandwidth can be to use a static flow rate per token. The number of tokens per flow is recalculated in given time intervals. This allows for a more accurate token management, in particular if flow rates change frequently. It also takes the mouse-elephant issue, a few large flows and many small flows, into account. However, it does not provide a better facility to forecast future flows. Since the flow rate of future flows is unknown, average values have to be used. This option also requires additional processing as well as memory resources. If the token mechanism is only used for routing, one token per flow and periodic recalculations of the average flow rate are sufficient. If the updated intervals are longer, fixed-bandwidth-fraction-tokens are more accurate since they are a simple measure of the used bandwidth. The price for the improved accuracy is additional resources.

V. Routing Strategies

This section gives quantitative estimates of how different gateway routing strategies influence performance. For upstream links two general cases can be identified: Links are equal (throughput, delay, cost); or Links can be prioritised. Possible routing strategies for a gateway include: Single link, failover, load balancing and overflow routing. The first three are common to many gateway appliances; the third strategy is unique to the Enterprise Overflow Router. *Failover* routes all traffic on the default interface, if this fails all traffic is routed on the backup interface. From a performance perspective failover and single link routing are equivalent, if link failures are not considered. Load balancing distributes flows randomly or in a round-robin fashion between available interfaces. Overflow routing uses the default link first and routes additional traffic on the secondary interfaces and so on.

If all upstream links are equal, load balancing is optimal. However, this is often not the case. Links can be prioritised,

for example, by cost or performance. In these cases load balancing will use lower priority links for more traffic than necessary. This either increases the cost or reduces the overall performance. Load-balancing is also unable to exactly spilt traffic which might lead to additional use of the links with high costs or bad performance. Overflow routing, on the other hand, will utilise the link with the highest performance metric first and still utilise all other resources, if required. Therefore, it naturally optimises the routing process.

Traditional routing has only taken one direction into account. This allows forward and reverse packets that belong to the same flow, to take different paths through networks. The scenario in this paper uses network address translation which ensures that forward and return packets are routed on the same interface. In practice, flows have two components, one upstream and one downstream and traffic is often asymmetric. Before a flow is routed, it is therefore necessary to make sure sufficient resources available in the forward, as well as reverse direction.

VI. Simulations

This section introduces exemplary simulations that underline the advantages of overflow routing in enterprise scenarios. The simulations emulate the network setup depicted in Figure 1. A local network is connected to the Internet via two links. Round-trip times of the links are set to 30 ms (ADSL) and 200 ms (wireless broadband) respectively. These links are also the bottle necks with 1.5 Mbit/s capacity each. The implementation uses the well known simulator NS2 with custom modifications to enable flow tracking and bandwidth estimation. The network is loaded with http traffic, based on the PackMime traffic model [11]. Each run simulates one hour, link loads are measured in 10 second intervals, after a one minute warm-up period.

Figure 2 depicts the loads on three links, a link (0) that captures all traffic and two links (1) (2) that connect the local network to the Internet. It shows the results for the overflow case and a traffic load of 5 new connections per second. The topmost line (bd0) depicts the load on the downstream path, which carries the combined load of link 1 (bd1 red/grey) and link 2 (bd2 yellow/light grey). The other three lines (bu0, bu1, bu2) show the load on the reverse paths in the same order. The colour coding is persistent between the different graphs. The operation of the overflow scheme is demonstrated: traffic is routed on the default link (bd1) and only when necessary, the overflow link is used. Most of the time, capacities are not fully utilised and the advantages of overflow routing are most visible in this cases. Figure 3 shows the graph for the same setup, loaded with the same traffic, however, loadbalancing is used as a routing scheme. The two alternative links are loaded evenly.

Figure 4 depicts a histogram of the *request response time* for completed http requests. Note that this is not the round-trip time and that the delay depends on available capacity as

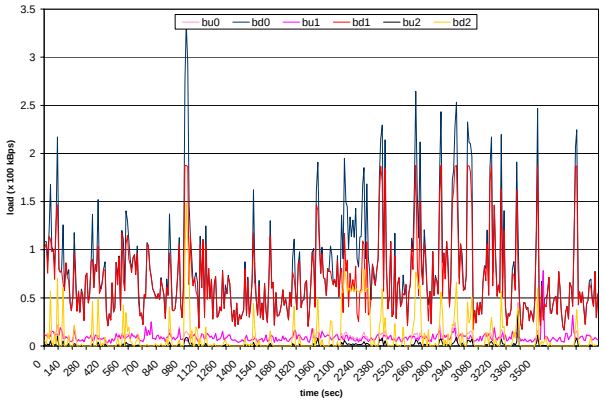

Fig. 2. Traffic - Overflow Routing (10 fl/s)

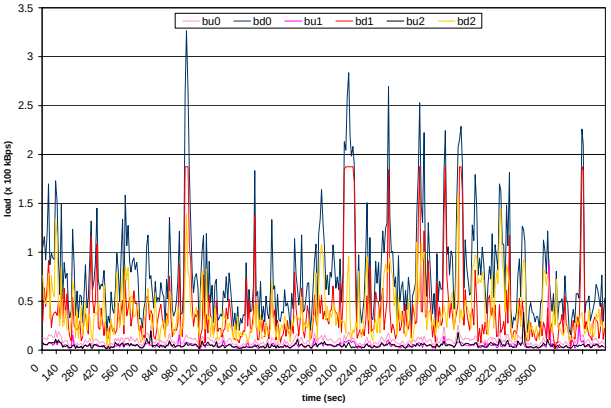

Fig. 3. Traffic - Loadbalancing (10 fl/s)

well as the request size. The x-axis shows delay bins and the y-axis shows the frequency count on a logarithmic scale. The median for overflow routing is 413 ms; for load balancing, it is 983 ms. It can be observed that for load balancing a large number of requests encounter the high latency on the second link. Overflow routing only uses this link when necessary and is therefore less penalised.

Figure 5 shows the overflow scenario at a higher load of 30 new connections per second. Figure 6 depicts results for the same traffic and overflow routing, however, the routing does not take both directions into account to make routing decisions. Since the bottleneck is on the downstream path, the traffic is never routed on the overflow link. It can be observed that the default link is used up to its downstream capacity limited. Figure 7 depicts request response delay histogram for both cases. The medians are 925 ms for bidirectional overflow routing and 12696 ms for unidirectional routing. Both links are be used in the bidirectional overflow scenario, however, only one link was used in the unidirectional scenario. Dynamic routing needs to route up- and down-stream traffic on the same path and take the load in both directions into account.

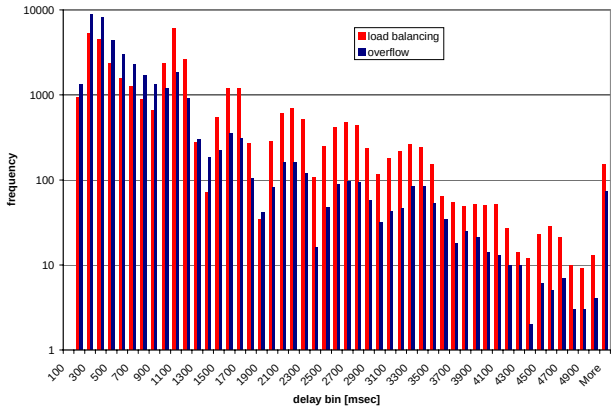

Fig. 4. Request Response Time - Loadbalancing & Overflow Routing

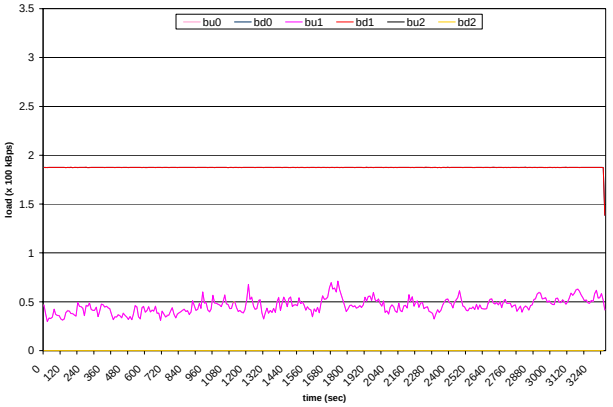

Fig. 6. Traffic - Unidirectional Overflow Routing (30 fl/sec)

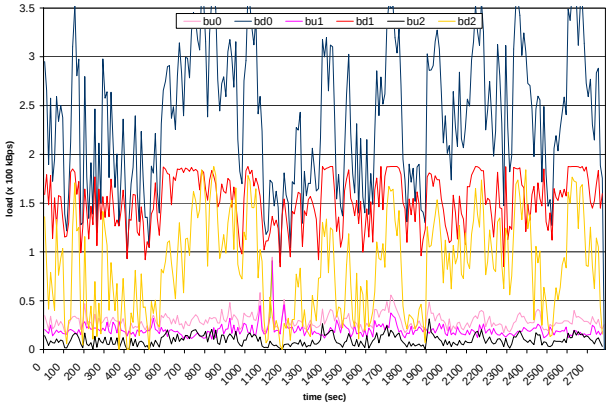

Fig. 5. Traffic - Overflow Routing (30 fl/sec)

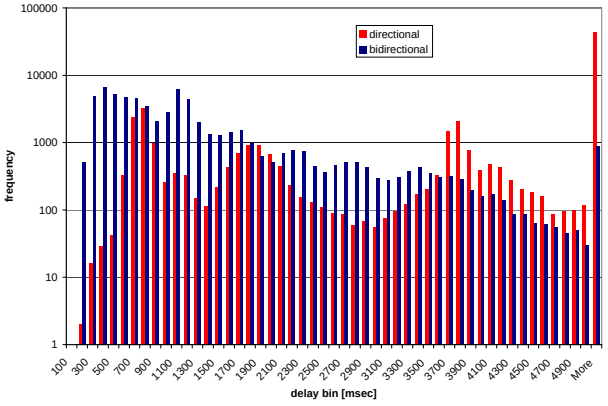

Fig. 7. Request Response Time - Uni & Bidirectional Overflow Routing

VII. Conclusion

One of the major challenges for dynamic routing in the Internet are various and changing flow rates. A token system that dynamically adapts the average flow rate is a simple tool allowing the forecast of link resources, required by dynamic routing. The proposed models have been implemented in an enterprise overflow router prototype that can operate at 100 Mbit/s line speed. It has been shown that overflow routing is superior to load balancing for multi-homed, NATed networks that do not use major router infrastructure. In particular exemplary simulations demonstrated how overflow routing naturally optimises link usage, benefiting performance and/or cost.

VIII. Acknowledgements

The author would like to thank the Australian Telecommunications Cooperative Research Centre (ATcrc) for their financial assistance of this work.

References

[1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. *Requirements for Traffic Engineering Over MPLS*. IETF, September 1999. RFC 2702.

[2] B. Fortz and M. Thorup. Optimizing OSPF/IS–IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, May 2002.

[3] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine*, 41(5):122–128, May 2003.

[4] I. Van Beijnum. *BGP: Building Reliable Networks with the Border Gateway Protocol*. O'Reilly, September 2002.

[5] G. R. Ash. *Dynamic Routing in Telecommunication Networks*. McGraw-Hill, 1997.

[6] J. Regnier, F. Bedard, J. Choquette, and A. Caron. Dynamically controlled routing in networks with non-DCR-compliant switches. *IEEE Communications Magazine*, pages 48 –52, July 1995.

[7] R.J. Gibbens, F.P. Kelly, and P.B. Key. Dynamic alternative routing - modelling and behaviour. *Proc 12th International Teletraffic Congress (ITC 12), Turin Italy*, 1988.

[8] K. Kawashima and A. Inoue. State- and time-dependent routing in the NTT network. *IEEE Communications Magazine*, pages 40–47, July 1995.

[9] A.A. Kist and R.J. Harris. Scheme for alternative packet overflow routing (SAPOR). *In IEEE Workshop on High Performance Switching and Routing (HPSR 2003), Turin, Italy*, June 2003.

[10] A.A. Kist, B. Lloyd-Smith, and R.J. Harris. A Simple IP Flow Blocking Model. *The 19th International Teletraffic Congress (ITC19), Beijing, China*, August 2005.

[11] J. Cao, W.S. Cleveland, Y. Gao, K. Jeffay, F.D. Smith, and M.C. Weigle. Stochastic models for generating synthetic http source traffic. *Proceedings of IEEE INFOCOM,Hong Kong,*, March 2004.